```
git init                        # Initialize repo
git config --global user.name "" # Set username
git config --global user.email ""# Set email
git add file.txt                # Stage file
git commit -m "message"         # Commit with message
git status                      # Show repo state
git log                         # View commit history
```

```
git clone https://github.com/user/repo.git      # Clone repo
git remote add origin https://github.com/user/repo.git # Link remote
git push -u origin main                         # Push to GitHub
git fetch origin                                # Download changes (no merge)
git pull origin main                            # Download & merge changes
```

```
git branch feature-1          # Create branch
git switch feature-1          # Switch to branch
git checkout feature-1        # Alternate switch (old)
git merge feature-1           # Merge into current branch
git rebase main               # Rebase onto main
git stash                     # Save uncommitted work
git stash apply               # Restore stash (keep stash)
git stash pop                 # Restore stash (remove stash)
```

```
git checkout -b feature/new-feature         # Create & switch to feature branch
git push -u origin feature/new-feature      # Push feature branch
# Then create PR on GitHub/GitLab UI
git checkout main
git pull origin main                        # Update main locally
git checkout feature/new-feature
git merge main                              # Merge main to resolve conflicts
# Fix conflicts, then:
git add <files>
git commit -m "Resolve merge conflicts"
git push origin feature/new-feature
```

```
git stash                # Save uncommitted changes
git stash list           # List all stashed changes
git stash apply stash@{0}  # Restore stash without deleting
git stash pop            # Restore stash and delete it
git rebase -i HEAD~3          # Interactive rebase last 3 commits
git commit --amend            # Amend last commit
git reset --soft HEAD~1        # Undo last commit, keep changes staged
git reset --hard HEAD~1        # Undo last commit, discard changes
git revert <commit-hash>       # Undo commit safely
git tag v1.0                  # Create tag
git push origin --tags         # Push all tags to remote

git log --oneline -n 5
git log -n 5 --pretty=format:"%h - %an, %ar : %s"

git checkout target-branch
git cherry-pick <commit-hash>
git cherry-pick <start_commit>^..<end_commit>
# If conflicts:
git add <resolved-files>
git cherry-pick –continue

git log --author="JohnDoe"
git log --since="YYYY-MM-DD"
git log --until="YYYY-MM-DD"
git log --author="JohnDoe" --since="YYYY-MM-DD" --until="YYYY-MM-DD"
```