```
** GETTING STARTED **

# start a new repository
git init

# clone an existing repository
git clone <url>


** PREPARE TO COMMIT **

# add a specific file to staging
git add <file>

# add all files to staging
git add .

# interactively choose what to stage
git add -p

# move/rename a file
git mv <old> <new>

# delete a file
git rm <file>

# remove file from git but keep locally
git rm --cached <file>

# unstage a file
git reset <file>

# unstage everything
git reset

# check current status
git status



** MAKE COMMITS **

# commit and open editor
git commit

# commit with message
git commit -m "message"

# commit all tracked changes
git commit -am "message"



** BRANCHING **

# switch branch
git switch <name>

# switch branch (older command)
git checkout <name>

# create and switch branch
git switch -c <name>
```

```
# create branch (older command)
git checkout -b <name>

# list branches
git branch

# list branches by recent activity
git branch --sort=-committerdate

# delete branch
git branch -d <name>

# force delete branch
git branch -D <name>


** DIFF CHANGES **

# show all changes
git diff HEAD

# show staged changes
git diff --staged

# show unstaged changes
git diff


** DIFF COMMITS **

# show commit diff
git show <commit>

# diff two commits
git diff <commit> <commit>

# diff file from commit
git diff <commit> <file>

# show diff summary
git diff <commit> --stat
git show <commit> --stat


** DISCARD CHANGES **

# discard unstaged changes in file
git restore <file>

# discard changes (older command)
git checkout <file>

# discard all changes in file
git restore --staged --worktree <file>

# discard changes (older command)
git checkout HEAD <file>

# reset everything (dangerous)
git reset --hard

# remove untracked files
git clean
```

```
# stash changes
git stash


** EDIT HISTORY **

# undo last commit (keep changes)
git reset HEAD^

# interactive rebase (squash commits)
git rebase -i HEAD~6

# show reflog history
git reflog BRANCHNAME

# reset to specific commit
git reset --hard <commit>

# amend last commit
git commit --amend


** HISTORY / LOGS **

# show log for branch
git log main

# show graph log
git log --graph main

# show compact log
git log --oneline

# show file history
git log <file>

# show file history including renames
git log --follow <file>

# search commits by text
git log -G "text"

# show who changed each line
git blame <file>



** MERGING & REBASING **


# rebase branch onto main
git switch banana
git rebase main

# merge branch into main
git switch main
git merge banana

# squash merge
git switch main
git merge --squash banana
git commit

# fast-forward merge
```

```
git switch main
git merge banana

# cherry-pick commit
git cherry-pick <commit>



** RESTORE FILE **


# restore file from commit
git checkout <commit> <file>

# restore file (modern)
git restore <file> --source <commit>



** REMOTES **


# add remote repository
git remote add <name> <url>



** PUSH **


# push main branch
git push origin main

# push current branch
git push

# push new branch
git push -u origin <name>

# force push safely
git push --force-with-lease

# push tags
git push --tags



** PULL / FETCH **

# fetch changes only
git fetch origin main

# pull with rebase
git pull --rebase

# pull and merge
git pull origin main

# pull default
git pull



** CONFIGURATION **
```

```
# set username
git config user.name "Your Name"

# set global config
git config --global user.name "Your Name"

# create alias
git config alias.st status

# view config manual
man git-config



** IMPORTANT FILES **

local git config file
.git/config

# global git config file
~/.gitconfig

# git ignore file
.gitignore
```